# The Official AMOS

## CLUB NEWSLETTER Vol 2 Issue 2

**Back for yet another issue !** Ignore what we said about AMOS 2 last issue, it doesn't exist ! It's actually called AMOS Professional, and Mandarin/ Europress will send the boys round to deal with anyone who calls it AMOS 2 !

Other than that, it will hopefully have all the nice new features we mentioned, and Chris Payne of Europress expects it to be ready at the end of this year.

**Tons of reviews** this issue. Everything seems to have been released at the same time ! We have Easy AMOS, the excellent tutorial version of AMOS, Stephen Hill's new book, the Amiga Games Makers Manual (Yes it really is out !) and our own CTEXT 2. There's also news of some more extensions about to be released. Maybe we should do a separate newsletter just for reviews ?!

**The Disk** Issue 1 was finally released a few weeks back, with just about everything on it. Not only were all the listings from Vol. 2 Issue 1's newsletter there, but also programs like the checksum checker, and de-coloniser, a load of 10-liners, and the full Galaxy Rons game ! Issue 2 (The one for this newsletter) will contain a program to generate Extension Libraries for machine coders, the full Shuffle 2 extension and plenty of new toys. It will also be released on time (within 2 weeks of this newsletter) as we've sorted out the programs that caused the delay in issue 1.

***Deja Vu* Moves**. Sandra has moved again. The new address for Deja Vu Software is:

**Deja Vu Software,**
**7 Hollingbrook,**
**Beech Hill,**
**Wigan.**
**WN6 7SG**

The telephone number stays the same: **0942 495261.**

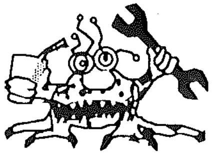
As of July 1st 1992, The Deja Vu Licenseware will become budget software and will be available in shops, as well as most P.D libraries. Because of the increased demand, Sandra will no longer be running the AMOS Club P.D library, which will now be run by Anne Tucker.

## In This Issue:

Absolute Beginners (Random Access Files) .......... 3
The Amiga Games Makers Manual (Book Review) .......... 7
Moving with Momentum .......... 8
Extension Update .......... 9
P.D & Licenseware news .......... 10
Shoot Out (Full game) .......... 12
Notes for Easy AMOS Users .......... 14
CTEXT 2 (Review) .......... 15
Sam Speed & The Case of Analog Joysticks (Extending AMOS) .... 16
Easy AMOS (Review) .......... 20

While the library changes over fully, Both Anne and Sandra will be handling P.D orders, but please be patient of any delays while the system changes.

# Bug Report



We had a new influx of bugs in Version 1.34 ! Due to Francois forgetting to remove his debugging routine, if you compile a program using **WAIT KEY**, and press the mouse button when it is waiting for the keypress, the machine will crash horendously ! Basically do this instead:

```
While Inkey$="" : Wend
```

**If you are using** screens over 409 pixels high, you'll find a bug with bobs, icons etc. They won't display past Y=409. It would seem to be some sort of bug with the Limit Bob system (it's been there all the time too !).

Both of these bugs will hopefully be fixed for AMOS Professional, although we will have to wait for it, and no doubt Francois will be able to add a few more for us !

## The AMOS Helpline ☎

As an AMOS Club member, you can call the AMOS Helpline at Shadow Software any time between 2p.m and 7p.m if you need help with your AMOS programming.
The Number is:
**0271-23544**
Remember to have your newsletter Vol. 2. issue 1 handy to quote your membership number.

# Useful Addreses

Here are some of the adresses to contact the various people in the AMOS world.

**Shadow Software/**
**The AMOS Club,**
1 Lower Moor,
Whiddon Valley,
Barnstaple,
N Devon. EX32 8NW
(AMOS Club offers and software)

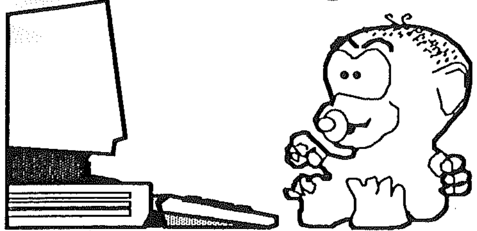
**The AMOS P.D. Library,**
1 Penmynydd Road,
Penlan,
Swansea,
SA5 7EH.
☎ 0792 588156
(AMOS P.D. & Totally AMOS disk Magazine)

**Europress/**
**Mandarin Software,**
Europa House,
Adlington Park,
Adlington, Macclesfield
Cheshire, SK10 4NP.
☎ 0625 859333
(AMOS to Easy AMOS upgrades and bug reports)

**Allen Computer Supplies**
432 Dorset Road, Croydon,
Victoria 3136
Australia.
☎ 03 725 3379
(Australian Deja Vu, TOME & AMOS Club distributer)

**David Lazerek,**
512 E 11th St,
Michigan City,
IN 46360. U.S.A
(American Deja Vu, TOME & AMOS Club distributer)

# Absolute Beginners



## Random Access Files

This issue, we are going to write a complete (if basic) quiz game, to show you how to use random access files.

Normally, data files are sequential, like the songs on a cassette tape, in that you have to read through all the data at the start of the file before you get to the bit you want.

Random access files are more like songs on a C.D player, in that you can jump directly to a particular **Record** and read it in, without having to go through the rest. Before going any further, I had better explain Files, Records & Fields:

**The File,** is the entire collection of:

**Records**, each of which is made up of:

**Fields**, each of which is one piece of data.

So for instance, in an address book file, Each record would be one persons data, with the fields of that record being the person's name, address and phone number.

For our Quiz game, we want a question, and three possible answers in each record (one record being one question). So our fields would be:

The Question
Answer 1
Answer 2
Answer 3

Obviously, only one of the answers would be correct (unless it is a Zen question !).

With Random Access files in AMOS, you have to tell AMOS how long each of the fields is going to be. So we have to decide how long our longest question will be, and how long our longest answer will be (The fields in one record are always the same length as in another record). In this game, the questions will be up to 160 characters long (4 Low resolution lines of text), and the answers will be up to 40 characters long.

So let's write a program that will put all the questions and answers onto the disk in a file called "Quiz.Dat"

First of all, we need to set up the screen and clear it, and set our record number to 1 (The start of the file)

```
10 Screen Open 0 , 320 , 200 , 16 , lowres
20 Paper 0 : Pen 1
30 Cls 0
40 Colour 1,$FFF
50 R=1
```

Now we need to open the random access file and define the field lengths. First open the file:

```
60 Open Random 1 , "Quiz.Dat"
```

The 1 is the channel number, so anything we do to channel number 1 will go to our file.

And define the fields (1 of 160 bytes and 3 of 40 bytes each). We will have Q$ for the question, and A1$, A2$ and A3$ for the answers...

```
70 Field 1,160 as Q$,40 as A1$,40 as A2$,40 as A3$
```

This will set up in channel 1 (our

file), 160 characters in Q$, and 40 each in A1$,A2$ and A3$.

Now we need to input the question...

```
80 Print "Enter the next Question"
90 Line Input "";Q$
```

Because the strings need to be exactly the length they were defined in the fields, we do the following to make sure:

```
100 Q$=Left$(Q$+Space$(160),160)
```

This makes Q$ the left hand 160 characters of the string you typed in, plus 160 spaces.

We can now input the 3 possible answers:

```
110 Line input "Answer 1 :";A1$
120 Line input "Answer 2 :";A2$
130 Line input "Answer 3 :";A3$
```

and make sure they are 40 characters long...

```
140 A1$=Left$(A1$+Space$(40),40)
150 A2$=Left$(A2$+Space$(40),40)
160 A2$=Left$(A2$+Space$(40),40)
```

To put them into the file, we use the **PUT chan,record** command.

```
170 Put 1,R
```

and add one to the record number, so that the next put will go to the next record.

```
180 R=R+1 : Rem you can also do Inc R
```

Now we just need to see if there's any more questions to be entered..

```
190 Print "Another Question Y/N ?"
200 A$=""  : Rem clear A$ to a Null String
210 While A$="" : Rem wait until it isn't empty
220 A$=Upper$(Inkey$)
225 Rem get a keypress
226 Rem and turn it to upper case
230 Wend : Rem end of the While/Wend loop
240 If A$="Y" Then   Goto 80
250 Close 1 : Rem closing the file
260 Rem finishes writing 265 rem to the file and must always
266 Rem be used when you have finished.
270 Print "Done !"
```

Those of you who complain that I should have used structured programming instead of a Goto to handle the loop can go back to lesson 1. You can't learn about structured code until you actually know what a goto does, it's the basic jump instruction !

Anyway, if you've typed in all the program lines above, you can **SAVE** the program (as Quizentry.Amos) and then Run it. Enter a few questions and answers such as **Q:**"How many programmers does it take to change a light bulb" **A1:** None, Lightbulbs are Hardware, **A2:** One, Most programmers can change lightbulbs, **A3:** Moped

The answer is obviously A1 (Those of you who thought it was A2 have never met a programmer, and anyone who thought it was A3 is a surrealist).

Type in the Questions without the "Q:" bit, and the answers without the "Ax:" bit (as they will be randomly displayed). The Third answer typed in will always be the correct one (although because they will be shuffled around, it won't always be the third answer).

Once you have a few questions and answers saved to disk, we can start on the game itself !

### The Game

The game program is quite similar to the question entry program, it just displays the questions & answers and waits for you to pick an answer. So we can open the screen and set up the colours exactly the same.

```
10 Screen Open 0 , 320 , 200 , 16 , lowres
20 Paper 0 : Pen 1
30 Cls 0
40 Colour 1,$FFF
```

We also need an array to hold the three answers (we can't just use 3 different variable names because the answers need to be shuffled). We also need an array to hold a flag to tell us which answer is correct. So:

```
50 Dim A$(3),F(3)
```

Now let's pick a random question

```
60 N=number of questions (You have to set this to the number of questions you typed in)
70 R=Rnd(N-1)+1 : Rem Pick a number from 1-N
```

Now we have to open the file exactly the same way as before:

```
80 Open Random 1 , "Quiz.Dat"
90 Field 1,160 as Q$,40 as A1$,40 as A2$,40 as A3$
```

and we can now read in the record...

```
100 Get 1,R
```

Q$ will now contain the question, and A1$,A2$ and A3$ will contain the answers. We now have to close the file...

```
110 Close 1
```

and shuffle the answers. To be able to do this, we have to transfer the answers to an array, and set the flag of the last one to 1 (to state that it is the correct answer).

```
120 A$(1)=A1$ : A$(2)=A2$ : A$(3)=A3$
130 F(1)=0 : F(2)=0 : F(3)=1
```

Shuffling 3 elements is quite simple..

```
140 For A=1 to 10
150 X=Rnd(1)+1
160 Swap A$(X),A$(3) : Swap F(X),F(3)
170 Next A
```

The swap command exchanges the two variables. Both the answer and the corresponding flag must be swapped, so that the same flag value stays with the answer.

We can now print up the Question and the three answers..

```
180 Print "Q: ";Q$
190 For A=1 To 3
200 Print "A";A;": ";A$(A)
210 Next A
220 Print "Pick one (1,2 or 3) :";
```

and wait for an answer

```
230 A$=""
240 While A$=""
250 A$=Upper$(Inkey$)
260 Wend
```

We simply check to see if it is in the range we want..

```
270 If (A$<"1") or (A$>"3") Then 230
```

```
280 Print A$
```

and we can check to see if they got it right !

```
290 A=Val(A$)
300 Print A$(A)
310 If F(A)=1
320 Print "Correct !"
330 Else
340 Print "Wrong !"
350 Endif
```

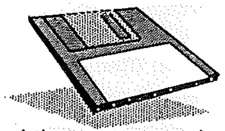
Note the use of an multi Line If..Else..Endif statement above. This works in the same way as If..Then, but it gives you more flexibility.

We can now finish the program off with a "Press a key to play again" routine, and a jump back to the start of the game..

```
360 Print "Press a key to play again"
370 Wait Key
380 Goto 70
```

There you go, a basic, very simple quiz game. Now see if you can expand it to have a points system...

# The Disk



If you hate typing in programs, or want advanced versions of the programs in this magazine, Issue 2 of the Disk has ALL the listings from this issue, and more....

This issue of The Disk contains: A Full version of Shoot Out, and the full quiz game with extra features.

The advanced version of the AMOS Club/Shuffle extension with 25 commands (plus the source code).

RAMOS 1.34

as well as other mini games and utilities...

You can order either a single issue of the disk for £2.00 (£2.50 overseas) or a subscription for £7.00 (£10.00 overseas) which gets you all the issues of the disk (including issue 1) for this year. The single disk fee for one disk is refundable on subscribing for the full set (Just tell us which one you don't need).

**The Disk is only available to AMOS Club members**, so include your membership number on the order form.

Please Send me...

☐ Issue 2 of The Disk for £2.00 (£2.50 overseas)

☐ A Subscription to The Disk for £7.00 (£10.00 overseas)

Name .................................................................... Membership # ................

Address ...........................................................................................................

.........................................................................................................................

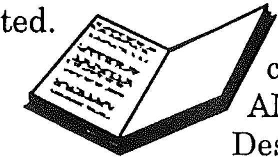
Postcode ..........................................................................................................

# The Amiga Games Maker's Manual

## By Stephen Hill.

Book Review by Aaron Fothergill

Stephen Hill's new book has been waited for since AMOS was first released. The ST version (to complement STOS) was a major success, and the Amiga Version for AMOS has been eagerly awaited.



Due to printing problems the publication of The Amiga Games Maker's Manual was delayed for quite a while, but it has finally been released (although my review copy has the glossary printed twice !).

The book covers the main aspects of writing games using AMOS and primarily covers arcade games (breakout & shoot-em-up styles), adventures and economic simulator (Kingdom) type games, with plenty of useful tips on all points of designing your games.

There are also chapters to cover using the AMOS Compiler, and AMOS TOME although because the text for the book was written before they were released, AMOS CText and AMOS 3D are not covered (disk based suplements are a posibility though).

The book is aimed at people who have some knowledge of Basic programming, but who haven't got to grips with AMOS yet, so you will need some form of programming experience, but the style and pace of the book suit the begining programmer quite well, and make it a suitable companion to an early AMOS user.

Despite the long delay in its release, the GM Manual is available now from bookshops, but make sure you ask for the AMOS Games Makers Manual, and not the STOS one !

**Conclusion**

Although it doesn't cover everything about AMOS (to do that would take around 2000 pages !), The Amiga Games Makers Manual is a worthwhile companion to any AMOS user.

**The Amiga Games Makers Manual**

**By Stephen Hill**
**400 pages**
**Published by Sigma Press**
**£16.95**

# Vectors Part 2 Moving with Momentum

A few issues back, I covered the use of vectors in a mini maths article. By popular demand, here's how to use the same ideas to give an object momentum !

In space, there is no drag on an object to slow it down. Once it is moving in a given direction, it stays moving. So if you had a vector of VX & VY, whatever other push you give to that object would be added to VX & VY. So:

```
Rem Vector Tester
Curs Off : Flash Off : Cls 0
Ink 1,1 : Bar 0,0 to 5,5
Get Bob 1,0,0 to 5,5
X#=160 : Y#=100 : VX#=0.0 :
VY#=0.0
Do
Bob 1,X#,Y#,1 : Wait Vbl
Rem Add Control to vectors
If Jleft(1) : VX#=VX#-0.2 :
Endif
If Jright(1) : VX#=VX#+0.2 :
Endif
If Jup(1) : VY#=VY#-0.2 :
Endif
If Jdown(1) : VY#=VY#+0.2 :
Endif
Rem Add Vectors
X#=X#+VX# : Y#=Y#+VY#
Rem Handle Wrap Around
If X#<0 : X#=X#+320 : Endif
If X#>319 : X#=X#-320 : Endif
If Y#<0 : Y#=Y#-200 : Endif
If Y#>199 : Y#=Y#+200 : Endif
Loop
```

This little tester allows you to control a bob around the screen with the joystick, note the use of floating point variables for X, Y, VX & VY. This allows smoother control.

This is all fine and wonderful in real space, but in games such as Asteroids and Thrust, you would get nowhere if your ship didn't slow down a little or get affected by gravity.

Gravity is simple, it is just a case of adding to the vertical vector VY. e.g

```
VY#=VY#+GR#
```

(Place this bit just after the controls, before the Add Vectors)

and set the value of GR# at the start of the program. Make it less than VY# is changed by the joystick, or you'll never be able to go upwards !

```
GR#=0.05
```

Adding drag to a vector is a little more difficult, but it isn't too bad ! Basically, you need to subtract a small drag factor from the vector. However, the drag factor must be the same sign as the vector. So we use the **SGN()** function to find out what direction the vector is. e.g.

```
DRX#=Sgn(VX#)*0.05
VX#=VX#-DRX#
DRY#=Sgn(VY#)*0.05
VY#=VY#-DRY#
```

Add this bit in with the gravity routine. The Drag can of course be altered by changing the **0.05** to a different value.

The only other bit you might want to do is a maximum speed. This is acheived by using the **Max/Min** functions to limit the values of VX# & VY# i.e.

```
VX#=Max(-16,Min(16,VX#))
VY#=Max(-16,Min(16,VY#))
```

This would go in just after the drag routine.

These little tips should help you write a more interesting game, something like Oids on the ST & Mac would be nice !

# Extension Update

**A lot has happened** extension wise since I published the last list. The list of officially logged extensions for AMOS is now up to number 15 !

As well as Rod Pascoe's new extensions (Extras & Disk Coms), we have two CDTV control extensions (both use the same number, as developers will want to choose between them), and a Sample player extension that can play samples direct from disk !

**The Current list:**

**1:** Music (Latest version on the 1.34 update includes Track Load/Play commands for Soundtracker files)
**2:** Picture Compactor
**3:** Request Extension
**4:** 3D Extension
**5:** Compiler Extension
**6:** Serial Extension
**7:** AMOS TOME (Current V3.2)
**8:** CText V1.32/V2.0
**9:** AMOS Club/Shuffle Extension
**10:** Dump Extension (Workbench 2 only at the moment)
**11:** CDTV/A570 Controller Extension
**12:** Extras Extension
**13:** Unused (Superstitious bunch !)
**14:** Disk Control Extension (Direct Track Read/Write, File Copy etc.)
**15:** DSam Sample Player Extension

**The two CDTV/A570 controller** extensions are both designed to give complete control of the CD Rom system, and are logged with Commodore by the respective developers.

**The P.D Screen** to Printer Dump extension sort of works, but only on Workbench 2.0 mostly. We're doing some work to get it working on a normal AMOS system.

**Rod Pascoe** has been working hard on two extensions, Extras and his new Disk extension. The **Extras extension** gives you the Execute command, so that you can run Workbench/Cli programs from within your AMOS program, a working Workbench Close & Workbench Open command (The original AMOS ones don't do anything !) and a couple of other goodies (like the Large and Small commands for changing screen update rate).

**The Disk extension** has commands to File Copy, Disk Copy, Format a Disk and Read/Write individual tracks/sectors. Watch this space !

**DSam is a new Sample player** extension, which gives you about 40 commands to manipulate and play samples. DSam also directly plays samples from Disk, up to 400K long at up to 10kHz (Faster rates coming soon). It's interrupt driven, so that it can be used at the same time as your program is running. I've seen one of the demos, and it looks great !

**CText** has just had a new version released (see the review in this issue), and **TOME Series IV** is under development, with a release probable in the next month or so. Series IV looks like having around 50 commands (14 more than V3.2) including automatic Map updating, Animating Tiles and Gravity Fall (For Boulderdash/Repton type games) commands built in !

**As usual** the AMOS Club/Shuffle extension is in a constant state of flux, with around 25 commands at the moment (such as Case Statements, Sam Speed and others). Some of the new commands are explained in the Programming Extensions article this issue, and the latest version of the extension is on this Issue of The Disk.

By the next issue, we'll probably be up to extension 20 and do you realise that over 200 commands have now been added to AMOS since it was released ?!

## P.D & Licenseware News

As you all know, Sandra Sharkey has been running Deja Vu Licenseware for almost a year, with a system of distribution that has provided an outlet for commercial software at a budget price, with a decent royalty going to the author. This system has worked well, with Licensed distributors (mostly P.D Libraries) duplicating and selling the software while paying the royalty through Deja Vu.

Unfortunately, the system had begun to be abusedand Sandra decided to change the system.

The new system makes all Deja Vu titles Budget Software. It is bought ready duplicated at a wholesale price by any retailer, and then sold on for any price up to £3.99. This means that royalties are collected for the authors when the disk is sold to the distributor (not 3 months after) and quality is better controlled (some distributors on the old scheme were still distributing old versions of software).

So how does this effect normal AMOS programmers and customers ? Well, you can still buy Deja Vu Budget Titles (There are about 100 now) from the major P.D libraries, as they will still be distributing them, but it also means you could see your titles in the shops (authors are encouraged to persuade their local shops to stock their titles). It should also encourage more authors to release their programs seriously, and gives a good outlet for AMOS programmers trying to sell their work.

The AMOS Club will be distributing Deja Vu Software, so you can order Deja Vu titles along with your AMOS Club software. Our price will be £3.50 per disk, with 50p postage and packing (£3.75 & £1.00 P&P overseas). We will be getting catalogues soon.

Because of the extra workload involved, Sandra has passed the AMOS P.D library on to Anne & Len Tucker, who will be running it from now on. The library has just reached the 400 disk mark (that's all AMOS titles) and new addititions include Easy AMOS versions of a lot of the popular AMOS routines disks, such as Treasure Search, Arcadia and Rainbow Warrior (Both AMOS & Easy AMOS versions of RW will be on issue 3 of "The Disk").

P.D prices are now £2.00 per disk, with 50p postage (European prices are the same, but £2.25 per disk for the rest of the world). A catalogue disk is available for £1.00 (which can be upgraded for 75p every time the catalogue is added to). The new address and phone number for the AMOS P.D library are on page 2, The new Deja Vu address is on page 1.

**Along with AMOS TOME, AMOS CTEXT is used by all the major AMOS programmers, and has been used in a number of commercial programs.**

This new version of CText is **only available to AMOS Club members** who have CText V1 and is only available direct from the AMOS Club (It will not be made available elsewhere).

CText V2.0 adds mulitple fonts, a much improved editor, diagonal and vertical text and much more to the Ctext System. In fact the command set has TRIPLED in size (16 commands now) since version 1.32.

CText V2.0 also comes with a 20 page manual, and 50 fonts on disk, as well as a program to automatically convert your Amiga Dos Fonts to Ctext format.

"I haven't used the Print or Text commands since the Ctext command was written" Len Tucker, Programmer and author of Totally AMOS magazine

CText V2.0 is only available from the AMOS Club by sending in your old CText V1.0 Disk with a Cheque/Postal Order/International Money Order for £10 made out to "The AMOS Club". (Use the order form below)

**AMOS TOME is currently being used by all the major AMOS programmers,** including Len Tucker, Aaron Fothergill, Peter Hickman, Phil South and more. Version 3.2 has 36 new commands for AMOS, and has **the most powerful map editor on any computer**.

Coming soon is TOME Series IV, which will be available as an upgrade toV3. This version is set to include automatic map updating, animating tiles and even a rock falling command, so that you can do Boulder Dash/Repton type games !

"Go and buy it now.." Phil South, Amiga Shopper Magazine.

"AMOS TOME has everthing in it, I'll never use another map system" Len Tucker, Programmer and author of Totally AMOS magazine.

Please Send Me:

| | |
|---|---|
| ☐ AMOS CText V2.0 (I enclose my old Ctext Disk) | **£10.00** |
| ☐ AMOS TOME V3.2 with free Goodies Disk 1 | **£19.99** |

Name .................................................. Membership # ..................

Address ..........................................................................................

.....................................................................................................

.................................................................. Postcode ...........................

Prices include U.K VAT and P&P. When ordering from overseas, please note that we can only accept cheques made out in pounds sterling and drawn on a U.K bank. (Eurocheques and American Post office money orders made out in pounds can also be accepted).

# SHOOT OUT !

By Aaron Fothergill

**A total, complete game here !** Everyone remembers the old Centipede game (and the strange tracker ball it used for control). Well here it is in full AMOSivision ! The game has been designed with a slight wild west theme, but you can use whatever graphics you want. You will need to create a sprite bank with the following bobs in it...

| Bob | Image |
|---|---|
| 1 | Players Gun |
| 2 | Bullet |
| 3 | Stagecoach frame 1 |
| 4 | Stagecoach frame 2 |
| 5 | Cactus |

All the bobs are 16x16 and have the hotspot at the bottom centre, except for the player's gun (Top Centre) and the bullet (Centre). The 3 objects with 2 frames animate between the two frames, so design accordingly. The stagecoach looks sort of like this:

and makes up the segments of the "centipede".

Once you have created your Bobs, save them, "New" the program, and load the bobs back in direct mode. Now you can start typing in the program (unless you have "The Disk", as it is already typed in for you). Remember not to type in the Line number & Bracket or the Checksum in square brackets. These are for use with the Checksum Checker program (issue 5/6 & on "The Disk" Issue 1).

```
 1) Dim C( 19, 13), WX( 19),
WY( 19), WDX( 19) [72]
 2) Screen Open 0, 320, 200,
16, Lowres : Curs Off : Flash
Off : Cls 0 [50]
 3) Screen Open 1, 320, 8,
16, Lowres : Curs Off : Flash
Off : Cls 0 : Autoback 0 [41]
 4) Screen Hide 1 [60]
 5) Screen 0 [20]
 6) Get Sprite Palette [34]
 7) Hide On [9]
 8) For A= 1 To 40 [40]
 9) X= Rnd( 19) : Y= Rnd(
8)+1 [6]
 10) If C( X, Y)= 0 [77]
 11) C( X, Y)= 1 [107]
 12) Paste Bob X*16, Y*16, 5
[41]
 13) End If [64]
 14) Next A [97]
 15) Double Buffer [26]
 16) T= 0 [89]
 17) X= 160 : Y= 190 [53]
 18) LVL= 1 [91]
 19) LIVES= 3 [75]
 20) Bob Update Off [49]
 21) While LIVES>0 [53]
 22) DEAD= 0 [14]
 23) BX= -1 [58]
 24) NB= 0 [5]
 25) Bob Off [8]
 26) SCOREO= -1 [39]
 27) While DEAD= 0 [122]
 28) If NB= 0 : Gosub
MAKEBADDIES : End If [93]
 29) Bob Clear [29]
 30) For A= 0 To 19 [37]
 31) If WX( A)>-1 [66]
 32) Bob A+1, WX( A), WY( A),
3+T [85]
 33) End If [64]
 34) Next A [97]
 35) Bob 0, X, Y, 1 [67]
 36) If BX>= 0 [14]
 37) Bob 21, BX, BY, 2 [124]
 38) End If [64]
 39) If SUP>0 [127]
 40) Screen Copy 1, 0, 0,
320, 8 To Logic( 0), 0, 192
[94]
 41) Dec SUP [52]
 42) End If [64]
 43) Bob Draw [78]
 44) Screen Swap [21]
 45) Wait Vbl [82]
 46) If Bob Col( 0, 1 To 20)
[52]
 47) DEAD= 1 : Boom [18]
 48) End If [64]
 49) X= X Screen( X Mouse) :
Y= Max( 144, Min( 176, Y
Screen( Y Mouse))) [36]
 50) If BX<0 and Mouse Key= 1
[72]
 51) BX= X : BY= Y : Shoot
[123]
 52) End If [64]
 53) If BX>= 0 [14]
 54) If Bob Col( 21, 1 To 20)
[17]
 55) Gosub HITBADDY [108]
 56) End If [64]
 57) BY= BY-12 [27]
 58) If BY<-4 : BX= -1 : Bob
Off 21 [40]
 59) Else [62]
 60) CBX= Min( 19, BX/16) :
CBY= Max( 0, BY/16) [56]
 61) C= C( CBX, CBY) [29]
 62) If C [104]
 63) C( CBX, CBY)= 0 [110]
 64) Cls 0, CBX*16, CBY*16 To
CBX*16+15, CBY*16+15 [80]
 65) BX= -1 : Bob Off 21 [17]
 66) End If [64]
 67) End If [64]
 68) End If [64]
 69) T= 1-T [32]
 70) For A= 0 To 19 [37]
 71) If Abs( WDX( A))<>8 [78]
 72) WXO= WX( A) [51]
 73) WX( A)= WX( A)+WDX( A)*4
[23]
 74) If( WX( A)<16 and WDX(
A)<0) or( WX( A)>308 and WDX(
A)>0) : WX( A)= WXO : WDX(
A)= -WDX( A) : WY( A)= WY(
A)+4 : End If [78]
 75) If WY( A)>192 : WY( A)=
16 : End If [80]
 76) If C( WX( A)/16, ( WY(
A)-4)/16) [88]
 77) WDX( A)= -WDX( A) : WY(
A)= WY( A)+8 : WX( A)= WXO
[39]
 78) End If [64]
 79) Else [62]
 80) If WX( A)>-1 [66]
 81) WY( A)= WY( A)+WDX( A)
[14]
 82) If WY( A)>210 [96]
 83) WX( A)= -1 : Bob Off 1+A
: Dec NB [91]
 84) If NB= 0 : SCORE=
SCORE+LVL*1000 : Inc LVL :
End If [62]
 85) End If [64]
 86) If WY( A)<32 and WDX(
A)= -8 : WDX( A)= -1 : End If
[94]
 87) End If [64]
 88) End If [64]
 89) Next A [97]
 90) If SCORE<>SCOREO : Gosub
UPSCORE : SCOREO= SCORE : End
If [74]
 91) Wend [57]
 92) Dec LIVES [38]
 93) Wend [57]
 94) End [107]
 95) UPSCORE: [127]
 96) S$= Right$(
"000000"+Mid$( Str$( SCORE),
2), 6) [92]
```

```
 97) Screen 1 [21]
 98) Ink 2, 0, 2 [93]
 99) Text 0, 7, ″SCORE:″+S$+″
LEVEL:″+Str$( LVL)+″
LIVES:″+Str$( LIVES) [43]
 100) SUP= 2 [88]
 101) Screen 0 [20]
 102) Return [42]
 103) HITBADDY: [52]
 104) For A= 0 To 19 [37]
 105) If Col( A+1) [53]
 106) If WDX( A)<8 [32]
 107) WDX( A)= 8 : Boom :
SCORE= SCORE+100 [39]
 108) If WY( A)>= 128+LVL*4 :
WDX( A)= -8 : End If [88]
 109) End If [64]
 110) BX= -1 : Bob Off 21
[17]
 111) A= 999 [68]
 112) End If [64]
 113) Next A [97]
 114) Return [42]
 115) MAKEBADDIES: [104]
 116) For A= 0 To 19 : WX(
A)= -1 : Next A [45]
 117) For A= 0 To Min( 19,
10+LVL) : WX( A)= 319-A*14 :
WY( A)= 16 : WDX( A)= -1 :
Inc NB : Next A [95]
 118) Bob Off [8]
 119) Bob Update [94]
 120) For A= 1 To 20 [46]
 121) X= Rnd( 19) : Y= Rnd(
8)+1 [6]
 122) If C( X, Y)= 0 [77]
 123) C( X, Y)= 1 [107]
 124) Paste Bob X*16, Y*16, 5
[41]
 125) End If [64]
 126) Next A [97]
 127) Return [42]
```

# Notes for Easy AMOS Users

With the release of Easy AMOS, we have had a lot of people join the AMOS Club who only have Easy AMOS and are planning on upgrading to AMOS after learning to program. Obviously, this means that some of the articles in the newsletter aren't relevant. However, programs such as Shoot Out, and articles such as the absolute beginners series should translate into Easy AMOS with no problems. Basically, try the program out, and if a command is displayed by Easy AMOS in uppercase rather than lowercase with the first character uppercase, you will have to find an equivalent command.

The AMOS P.D Library is currently compiling disks of Easy AMOS programs converted from the P.D disks of AMOS programs. Games such as Treasure Search and Reversi have been converted. They also have an Easy AMOS version of the RAMOS disk so that you can run your program without Easy AMOS. Contact Anne Tucker on 0792 588156 for info.

Remember also that if you get stuck or need a little help, you can always call us on the AMOS Helpline:

☎ 0271 23544 (Between 2pm & 7pm is the best time).

**Oddessey BBS is a new part time Bulletin Board run by Ben Ashley. It is open between 12 midnight & 7am and can be contacted on 0533 705630.**

# Ctext$^2$

By Shadow Software.
Reviewed by Len Tucker.

This is an overview of the long awaited CTEXT 2 by Aaron Fothergill of Shadow Software, written by its Number One Fan!

If you thought that Ctext was the ultimate Text extension, then you were wrong, CTEXT 2 gives you even more flexibility for displaying text.

This new version of Ctext comes with its own printed manual which will be a great help to users as the instructions are readily available in front of you as you get to grips with the inner workings of the program. To show off CTEXT's abilities, little demo programs are included on the disk for you to look at. Installing CTEXT 2 is just a matter of following the instructions in the manual, so you shouldn't have any problems. There are several new commands - all of them are essential, some of them are amazing! There is a new version of the Font Setter program which is more User Friendly than the original. Using this program is very straightforward, basically it allows you to proportionally space your font and customise it ready for Ctext. Another useful addition to Ctext 2 is a utility program which converts AmigaDos fonts to Ctext fonts which will save a lot of time.

Ctext 2 contains all the original commands of Ctext, in addition here are a couple of the new commands:-

1. **Font n** - This command allows you to select which font you want to use - yes, Ctext now supports multiple fonts which must make it the best addition to this version. So, if memory allows, you can have 4 x 64 character fonts in memory at the same time and toggle between them.

2. **Font Step n** - This is similar to Font Shift but affects the vertical axis instead of the horizontal. In the right hands, the effects of this command have to be seen to be believed !

3. **=Cfont$()** - This allows you to add the font you wish to use to a string itself. eg A$ = Cfont$(0) + “AMOS” + Cfont$(1) + “is Amazing!”

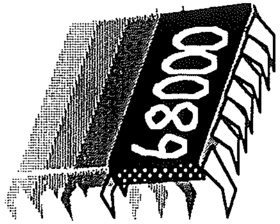
The Ctext System now has 16 commands which is quite a large number considering that it deals only with Text. Another addition is Stencil Fonts, this means that you can now change the colour of your font in run time, which will add variety to your programs.

**General Remarks.**

I am unashamedly the Number 1 Fan of Ctext! The first version was great, but now the things that were missing from it have appeared in Ctext 2. I have used Ctext both in Commercial and Licenseware titles as it saves so much time and is so effective. Using it will give a more professional appearance to your programs and make your programming life a lot easier. Just a note for those of you who write for P.D., Licenseware or Commercial releases. If you use Ctext (1 or 2) then your program MUST be compiled, on no account must a program containing Ctext go out uncompiled. This is a condition of its use, not a limitation of the program.

**CTEXT 2 is only available to AMOS Club Members and costs £10 as an upgrade to CTEXT. Send in your old CTEXT disk, and £10 to Shadow Software at the usual address. Remember to include your membership number**

# Sam Speed & The Case of Analog Joysticks



Extending AMOS Yet again !

Way (way way) back in issue 5/6 I showed you the AMOS Shuffle extension which was all of 3 commands long ! This gave you an introduction to programming extensions for AMOS using machine code.

This issue we add 3 sets of commands, SAM SPEED to control the playback speed of a sample (great for engine noises), CASE and OF statements (for Pascal freaks) and Analog Joystick commands for use with Amiga Analog joysticks.

As with issue 5/6, you'll need to use Devpak or a similar assembler to handle to code, which should be added to the current shuffle extension (The comments show where the bits should be inserted).

The @ characters show where you should hit the Tab key.

For the terminally lazy (or the non coders) the full shuffle 2 extension is on this issue of "The Disk" in both source code and extension form.

## Using the Commands

**SAM SPEED chan,speed**

chan is a bitmap (0-15) of the channels to be effected, speed is the playback speed in hertz. e.g

```
Sam Speed 15,10000
```

would change the speed of the samples on all 4 channels to 10000 hertz.

Because Sam Speed uses the sound chip directly, you can use it to change the rate of any sound.

**CASE OF n**

**CASE OF n$**

These 2 commands (one for integers, one for strings) are the start of a CASE statement. Basically CASE takes a note of the value of n or n$ for future checks with the OF command.

**OF val,index**

**OF val$,index**

The OF command compares the noted value and compares it with val (or val$). If it is equal, it put the value Index into a reserved variable called CASE.

**=CASE**

This returns the index value from the OF commands.

An example of a CASE statement.

```
Do
A$=Inkey$
Case Of A$
Of "A",1
Of "B",2
Of "C",3
On Case Gosub A,B,C
Loop
A: Print "A pressed" : Return
B: Print "B pressed" : Return
C: Print "C pressed" : Return
```

**ANALOG SCAN**

This command scans the analog joysticks to find their current values. This should normally be done 1 vbl previous to actually reading the sticks.

**=ANALOG X(n)**

Returns the X co-ordinate (0-255) of analog stick n (n ranges from 0-1)

**=ANALOG Y(n)**

Returns the Y co-ordinate (0-255) of analog stick n.

```
* Add this bit in at the
start of your Shuffle
Extension
Dload@ MACRO
@ move.l @
ExtAdr+ExtNb*16(a5),\1
@ ENDM
Case @ equ @ 4
ICase @ equ @ 8
SCase @ equ @ 12
NB @ equ @ 16
NBO @ equ @ 20
* Then change the offset
table to look like this
X_Off   @ dc.w (L1-L0)/2,(L2-
L1)/2,(L3-L2)/2,(L4-L3)/2
@ dc.w (L5-L4)/2,(L6-L5)/
2,(L7-L6)/2,(L8-L7)/2
@ dc.w (L9-L8)/2,(L10-L9)/
2,(L11-L10)/2,(L12-L11)/2
@ dc.w (L13-L12)/2,(L14-L13)/
2,(L15-L14)/2,(L16-L15)/2
@ dc.w (L17-L16)/2,(L18-L17)/
2,(L19-L18)/2,(L20-L19)/2
@ dc.w (L21-L20)/2,(L22-L21)/
2,(L23-L22)/2
* AMOS Shuffle Extension
token table modifications
* Insert these at the end of
your token table
@ dc.w @ L_CASE3,-1
@ dc.b @
"case","$"+$80,"I2",-1
@ dc.w @ L_CASE1,-1
@ dc.b @
"!cas","e"+$80,"I0",-2
@ dc.w @ -1,L_CASE4
@ dc.b @ $80,"0",-1
@ dc.w @ L_OF3,-1
@ dc.b @
"of","$"+$80,"I2,0",-1
@ dc.w @ L_OF1,-1
@ dc.b @ "o","f"+$80,"I0,0",-
1
@ dc.w @ L_ANSCAN,-1
```

```
@ dc.b @ "analog
sca","n"+$80,"I",-1
@ dc.w @ -1,L_ANX
@ dc.b @ "analog
","x"+$80,"00",-1
@ dc.w @ -1,L_ANY
@ dc.b @ "analog
","y"+$80,"00",-1
@ dc.w @ L_SamSpeed,-1
@ dc.b @ "sam
spee","d"+$80,"I0,0",-1
*******************************
*@ Start of library
X_Lib
*@ COLD START
* This replaces the old L0
(it works out the clock rate)
L0
@ lea@ Table(pc),a3
@ move.l@
a3,ExtAdr+ExtNb*16(a5)
; 50/60 herz?
@ move.l@ #3546895,MusClock-
Table(a3)
@ EcCall@ NTSC@ @ * Is system
NTSC?
@ tst.w@ d1
@ beq.s@ ItsPAL
@ move.l@ #3579545,MusClock-
Table(a3)
; Go on!
ItsPAL
@ moveq@ #ExtNb,d0
@ rts
*
Table: @ dc.l @ 0
@ dc.l @ 0 @ ; CASE variable
@ dc.l @ 0 @ ; Case Integer
checker
@ dc.l @ 0 @ ; Case String
checker
@ dc.l @ 0 @ ; NB
@ dc.l @ 0 @ ; NBO
MusClock: @ dc.l @ 3546895
```

```
L1
* SAM SPEED c,s
* Change speed of samples on
channels C (bitmap) to speed
s
L_SamSpeed @ equ @ 1
@ move.l @ (a3)+,d3
@ move.l @ (a3)+,d0
@ Dload @ a0
@ move.l @ MusClock-
Table(a0),d4
@ divu @ d3,d4
@ moveq.l @ #0,d1
@ btst @ d1,d0
@ beq @ C2
@ move.w @ d4,$DFF0A6
C2: @ moveq.l @ #1,d1
@ btst @ d1,d0
@ beq @ C3
@ move.w @ d4,$DFF0B6
C3: @ moveq.l @ #2,d1
@ btst @ d1,d0
@ beq @ C4
@ move.w @ d4,$DFF0C6
C4: @ moveq.l @ #3,d1
@ btst @ d1,d0
@ beq @ ESS
@ move.w @ d4,$DFF0D6
ESS: @
@ rts
*
L8 These libraries are
L9 Reserved for Number/
L10 String conversion
L11 Commands, next issue
L12
L13
L14
L_CASE1 @ equ @ 14
* Case Command for Integers
@ Dload @ a0
@ move.l @ (a3)+,d0
@ move.l @ d0,ICase(a0)
@ move.l @ #0,Case(a0)
@ rts
L15
L_CASE3 @ equ @ 15
* Case Command for Strings
@ Dload @ a0
@ move.l @ (a3)+,a1
@ move.w @ (a1),d0
@ beq @ Nostring
@ move.l @ a1,SCase(a0)
@ move.l @ #0,Case(a0)
@ rts
Nostring:
@ move.l @ #0,SCase(a0)
@ move.l @ #0,Case(a0)
@ rts
L16
L_CASE4 @ equ @ 16
* Case Reserved Variable
@ Dload @ a0
@ move.l @ Case(a0),d3
@ move.l @ #0,d2
@ rts
L17
L_OF1 @ equ @ 17
* Case, OF Integer command
@ Dload @ a0
@ move.l @ (a3)+,d1
@ move.l @ (a3)+,d0
@ cmp.l @ ICase(a0),d0
@ beq @ YCase1
@ rts
YCase1: @ move.l @
d1,Case(a0)
@ rts
L18
L_OF3 @ equ @ 18
* Case, OF String command
@ Dload @ a0
@ move.l @ (a3)+,d1
@ move.l @ (a3)+,a1
@ move.l @ SCase(a0),a2
@ beq @ Nocase
@ move.w @ (a2),d2
@ move.w @ (a1),d3
@ cmp.w @ d2,d3
```

```
@ bne @ Nocase
@ move.l @ d2,d0
LOF3L: @ subq.l @ #1,d0
@ move.b @ 2(a1,d0),d2
@ move.b @ 2(a2,d0),d3
@ cmp.b @ d2,d3
@ bne @ Nocase
@ tst.l @ d0
@ bne @ LOF3L
@ move.l @ d1,Case(a0)
Nocase @ rts
L19
L_ANSCAN @ equ @ 19
@ move.l @ #$DFF000,a0
@ move.w @ #1,$34(a0)
@ rts
L20
L_ANX @ equ @ 20
@ move.l @ #$DFF000,a0
@ move.l @ (a3)+,d0
@ andi.l @ #$1,d0
@ asl.l @ #1,d0
@ addi.l @ #$12,d0
@ move.w @ 0(a0,d0),d3
@ andi.l @ #$FF,d3
@ move.l @ #0,d2
@ rts
L21
L_ANY @ equ @ 21
@ move.l @ #$DFF000,a0
@ move.l @ (a3)+,d0
@ andi.l @ #$1,d0
@ asl.l @ #1,d0
@ addi.l @ #$12,d0
@ move.w @ 0(a0,d0),d3
@ asr.l @ #8,d3
@ andi.l @ #$FF,d3
@ move.l @ #0,d2
@ rts
L22 @ moveq @ #0,d1
@ moveq @ #ExtNb,d2
@ moveq @ #-1,d3
@ Rjmp @ L_ErrorExt
L23
* @ Welcome message
X_Title @ dc.b @ 31,"AMOS
Club Extension V2.1",0
@ even
**************************************************************
X_End @ dc.w @ 0
* end the extension with a 0
*
```

Once you have modified the extension, assemble it as RANGE.Lib and log it in as extension #9.

Next issue, we'll add the MKB$, MKI$, MKL$ , CVB, CVI and CVL functions to Shuffle.

The version of Shuffle on this issue of "The Disk" includes all these commands and more, and you don't have to type it in ! It also contains a wiring diagram for the Analog Sticks.

By the way, a strange fact about Amiga Analog Joysticks. The fire button is normally wired to the right stick switch, which means you normally use =Jright(n) to detect if fire has been pressed. Also because of the way they are wired, you can press all the other directional switches into service, along with the fire button, and have FIVE fire buttons per joystick !

**I hope everybody likes the new look of the newsletter. Don't worry, we'll still be cramming as much programming in as possible, rather than just making it look nice !**

This Newsletter was written by and remains © Aaron Fothergill except for stated articles for which copyright remains with the original author. No Part of this publication may be copied in any form, without prior consent by Aaron Fothergill.

The AMOS Newsletter was put together on an Apple Macintosh using Aldus PageMaker.
Graphics by Adam Fothergill.
Printed by Creative Printers of Devon,
104 Boutport St., Barnstaple, N.Devon.
☎ (0271) 77655

# *Easy* AMOS

Review by Aaron Fothergill: **Since AMOS was first released**, there have been users who have fallen into the trap of thinking that it was a games creator, have bought it and found that they have no idea of how to program it.

Unfortunately, the AMOS manual is more of a reference manual than a tutorial and so Easy AMOS has been released to bridge the gap between the absolute beginners and the power users.

Easy AMOS starts off well, it boots up to an installation program that will put a personalised copy of the program onto your own disks, so that you can't damage your Easy AMOS disk. You can only use your own Easy AMOS backup, the actual program disks are only installers. Europress have been thoughtful enough to provide you with labels for your newly created Easy AMOS disks.

The easy AMOS editor is more user friendly than the AMOS variant, several of the utility programs are directly accessed from this menu. These utilities are cut back, easier to use versions of the full AMOS programs, such as the Sprite Editor. The Easy AMOS Image editor looks nicer than the AMOS one, and can handle larger areas, but lacks several functions that would have been useful for the beginner (such as the ability to automatically remove waste space around an image). The program is easy enough and powerful enough for most applications though.

The Easy AMOS Language is basically a cut back version of AMOS. The idea being that all the commands that distracted the learner from actually learning to program properly have been removed (such as AMAL) and only the more learnable ones left in. Apart from a few mistakes such as leaving in the disk font system, and leaving out commands such as =Jleft, =Jright, =Jup & =Jdown leaving you with =Joy and forgetting useful reserved variables such as Screen Height & Screen width, forcing you to do a `SCRW=Deek(Screen Base+76)` and `SCRH=Deek(Screen Base+78)`, this variant of the language works well.

Easy AMOS has a Tutorial manual, which works you through the language. Combined with the built in Quiz program (which tests what you know of Easy AMOS), this makes Easy AMOS a very good way to learn programming. The manual is easy to read, and although it doesn't take the user through the language in the easiest order, most non programmers should get to grips with it within a few hours. We tested the program on a few non-programmers, who were up and running with it quite rapidly.

In all, despite a few minor inadequacies, Easy AMOS is to be recommended as a tutorial for AMOS, and is well worth buying, especially as Europress Software have a special offer for AMOS Users. Just send in your AMOS Programs Master disk (The one with the Red Label), with £15 and they will send your disk back, along with Easy AMOS. This means that you are getting a tutorial for AMOS for £15 !

**Easy AMOS**

Europress Software

£34.99 (£15 for existing AMOS Users)